

**REMARKS**

Claims 1, 2, 5-12, and 15-20 stand rejected as unpatentable over *Schneck* (US 5,933,498) in view of *Alexander* (US 6,134,593), in view of *Slivka* (US 6,049,671), and further in view of applicant's own admissions (AOA). The applicants respectfully traverse that rejection.

The rejection cites passages from *Schneck* said to disclose elements of the rejected claims. The undersigned respectfully submits that the reference, at those passages or otherwise, does not support the teachings attributed to them. First of all, *Schneck* discloses a system for controlling access to data, but does not disclose a method for permitting the controlled installation of a software product onto a local machine, the method defined by Claim 1. Thus, although *Schneck* is concerned with controlling access to data, the reference is silent as to installing a software product on a local machine.

The rejection of Claim 1 asserts that *Schneck* discloses:

*"generating an installer identifier at the local machine based on the software product in response to a request to install the software product on the local machine."*

Figs. 1-3 and Items 127, 128, and 130 are cited in support of that alleged disclosure. However, Fig. 3 merely shows the access rules 116 relating to the data 106 being distributed to the user, but does not show generating an installer identifier as limited in that step. Those access rules may be encrypted and packaged with the data, or may be provided separately to the user (column 10, lines 34-37). The user can then access the data according to the rules provided therewith or provided separately (column 17, lines 34-37). For that access, a hardware-based access mechanism 114 (column 15,

lines 19 plus; column 18, lines 11-18) determines whether or not the requested data is protected. If it is protected, then access is permitted according to the access rules provided therewith or separately (column 18, lines 32-36 and 44-47). Otherwise, the requested data access is denied.

Nowhere does *Schneck* disclose or suggest "generating an installer identifier at the local machine based on the software product in response to a request to install the software product on the local machine". *Schneck* does not install software, does not generate an installer identifier at the local (i.e., user's) machine based on the software product or otherwise, and that reference thus does not disclose the step attributed to it.

*Schneck* does discuss enabling the creator of executable software to restrict the use of software to only those who have acquired permissions for its capabilities; see column 29, lines 26-46. That executable software is distributed in encrypted form and externally treated as data, as described above and at column 29, lines 30-32. However, the unencrypted executable code is kept wholly within the security perimeter of the hardware-based access mechanism 114 for execution (column 29, lines 43-46). That application of *Schneck* fails to teach "generating an installer identifier of the local machine...", as required by the software-installation method of Claim 1.

*Schneck* does mention that users may download executable software and then separately purchase rights to use that software (column 29, lines 48-57). However, that reference discusses only the active control of executable software (column 30, lines 15-28; Fig. 16), wherein authorization may be requested and granted/denied for functions of the software. *Schneck* does not discuss a method for permitting controlled *installation* of

a software product, does not disclose generating an installer identifier as recited in Claim 1, and would have no use for that undisclosed element.

Because *Schneck* relies on a hardware-based decryption and access mechanism 114, that reference need not concern itself with a software-based method for permitting controlled installation as disclosed and claimed by the present applicants.

The rejection of Claim 1 also states that *Schneck* discloses:

*"the generated installer identifier represents a characteristic of the software product medium on which the software product is stored."*

Figs. 10(a-b) and columns 17-20 are cited in support. As pointed out above, *Schneck* does not generate an installer identifier at a local machine. The "Accessing Operation" discussed at columns 17-20 merely checks to see whether access rules are available (either packaged with the data or already present in the hardware-based access mechanism 114, and then permits or denies access to the data according to those stored rules (column 20, lines 9-25). In other words, *Schneck*'s system receives a requested operation, determines whether or not that operation is permitted by the rules, and either permits or denies the requested access. This operation of *Schneck* does not meet the requirement of "comparing the generated installer identifier to a stored installer identifier on the software product", as required by Claim 1.

Claim 1 also requires:

*"storing a license file and a hardware identifier identifying the local machine on the local machine in response to a match between the generated installer identifier and the stored installer identifier."*

Column 22, line 51-column 24, line 38 of *Schneck* are cited to disclose this limitation.

*Schneck*, as pointed out above, does not disclose storing anything identifying a local machine, because he is not concerned with *installing* a software product on a local machine. (Scheck's discussion of downloading and operating executable software, at column 29, line 48-column 30, line 28, discloses only operational control of executable software.) That disclosure is not concerned with permitting controlled installation of a software product, and that passage contains only a general discussion (column 29, lines 34-46) of acquiring a license to execute software. *Schneck* relies on a hardware-based access mechanism 114 to control decryption and protection of executable code, which is kept wholly within the security perimeter of that access mechanism 114 for execution (column 29, lines 43-46).

Accordingly, *Schneck* fails to disclose or teach "storing a license file and a hardware identifier...". Whether or not *Schneck's* "Rules" are deemed equivalent to the applicants' "license", as the rejection suggests, *Schneck* still fails to disclose or teach the foregoing limitation required by the method of Claim 1.

The rejection next states that *Schneck* discloses:

*enabling a complete installation of the software product on the local machine including installing at least one run-time file needed to execute the software product, in response to the match between the generated installer identifier and the stored installer identifier*, citing column 30, lines 6-47 as well as previous citations from *Schneck*.

Once again, *Schneck* does not disclose installing a run-time file in response to a match between a (not disclosed) generated installer identifier and a (also not disclosed) stored installer identifier. Column 30, lines 6-47 discuss using *Schneck's* invention for control of executable software (lines 15-16), but not for installing such software on a local machine. Although executable software in unencrypted form (within the hardware-based secure access mechanism 114) is executable and presumably includes a run-time file for that purpose, the reference fails to teach or suggest installing that run-time file at all, and certainly not "in response to [a] match between [a] generated installer identifier and [a] stored installer identifier" as required by Claim 1.

The rejection of Claim 1 concludes by saying the foregoing citations from *Schneck* disclose:

*"whereby the stored license file is associated only with the software product installed on the local machine and the hardware identifier is associated only with the local machine and the stored license file can be subsequently accessed to enable the execution of the completely installed software product including the installed at least one run-time file on the local machine but cannot be used with a separate software product or to execute the software product on a machine other than the local machine."*

As mentioned above, *Schneck* does not disclose a "stored license file" —by whatever name— associated only with a software product installed on a local machine. That reference also does not disclose a "hardware identifier" associated only with the local machine. Further yet, *Schneck* does not address the issue of generating and storing a license file to enable execution of completely installed software only on a local machine

but not with another machine, nor with a separate software product. These limitations simply do not occur in *Schneck*, as that reference deals with other aspects of software management, namely, controlling access to data or software based on permissions acquired for various levels of access.

The rejection asserts that *Schneck* discloses utilizing the hardware serial number for generating a key, which would have made obvious the applicants' "storing... a hardware identifier identifying the local machine on the local machine...". However, *Schneck* there discloses obtaining and validating the system serial number to calculate the rule-encrypting key  $K_D$  as a function of the validated serial number for some appropriate function, for example, an enquiry to obtain the public key so as to ensure that the serial number is authentic (column 14, lines 36-42). This disclosure is a part of *Schneck's* "Authoring Mechanism", at which time the data-encrypting key  $K_D$  is created for a particular data set. That encrypted data key  $K_D$  and the encrypted rules are then stored as packaged rules 152 for subsequent distribution (column 14, lines 47-48). This disclosure by *Schneck* concerns only the creation of encrypted rules for packaging and subsequent distribution. It does not disclose any aspect of installing a software product on a local machine.

The rejection recognizes that *Schneck* does not specifically disclose that his teachings may be applied to software installation on a local machine, as claimed herein. However, *Alexander* is cited as directed to installing software product onto client machines and is asserted to disclose those features. Although *Alexander* does indeed discuss installing a software application, it is not seen how one of ordinary skill would find it obvious to combine those teachings with the different teachings of *Schneck*. The

rejection appears to assume obviousness "because software is a subset of digital property". With respect, that observation would support combining any techniques involving subsets of digital property, a conclusion which far exceeds the requirement that an asserted combination of references must be obvious to one of ordinary skill, based on the teachings of the references. Further, *Alexander* teaches a method including storing an installation identifier on a local machine such that the installation identifier, once defined, may be used many times for separate products from a given vendor (column 4, lines 37-39). That teaching by *Alexander* is contrary to the invention defined in Claim 1, wherein "... the stored license file is associated only with the software product installed on the local machine and the hardware identifier is associated only with the local machine". Accordingly, even assuming *arguendo* the hypothetical combination of *Alexander* with *Schneck*, one of ordinary skill still fails to find at least one element required by the combination of Claim 1.

*Slivka* is cited only for disclosing a hashing algorithm and associated steps. That reference fails to teach or suggest a method including installation of a hardware identifier identifying the particular machine on which the software product and its license file is to be stored.

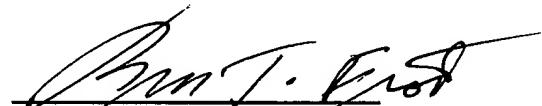
Summarizing the foregoing, the software installation method of Claims 1 et al. would not have been obvious to one of ordinary skill in the art in view of *Schneck*, *Alexander*, and *Slivka*. Those references fall well short of teaching or suggesting many elements required by Claim 1. One of ordinary skill, based only on those teachings and without knowledge imparted by the present applicants, would not have known those

elements nor the overall combination of the present invention. Claims 1 et al. are, accordingly, patentable over those references.

The foregoing is submitted as a complete response to the Office action identified above. The undersigned respectfully submits that the application is in condition for allowance and solicits a notice to that effect.

Respectfully submitted,

MERCHANT & GOULD



Date: January 23, 2006

Roger T. Frost  
Reg. No. 22,176

Merchant & Gould, LLC  
P.O. Box 2903  
Minneapolis, MN 55402-0903  
Telephone: 404.954.5100

